ORIGINAL ARTICLE

INTERNATIONAL JOURNAL OF ARTIFICIAL INTELLIGENCE

ISSN: 2692-5206, Impact Factor: 12,23
American Academic publishers, volume 05, issue 09,2025

Journal: https://www.academicpublishers.org/journals/index.php/ijai

PREPARING A DATASET FOR FACE DETECTION IN COMPUTER VISION

Rustamov Ilhom A'zam ugli
Diplomat University intern teacher
ilhomsaep0124@gmail.com
Khojamurotov Azizbek Shukhrat ugli
Tashkent University of Information Technologies named after Muhammad Al-Khwarizmi
azizbekxojamurotov2001@gmail.com

ABSTRACT: In recent years, the rapid development of artificial intelligence and computer vision technologies has significantly increased interest in face detection tasks. The effectiveness of face detection systems largely depends on the quality of the datasets used for training. Therefore, preparing a high-quality and diverse dataset has become a crucial scientific and practical challenge. This article provides a detailed overview of the dataset preparation process for face detection. It examines essential stages such as data collection and cleaning, face annotation, image normalization, and augmentation. These steps ensure dataset diversity and balance, which in turn contribute to the accuracy and robustness of face detection models.
The findings highlight that a well-prepared dataset not only improves the overall performance of face detection but also ensures that models remain reliable under various conditions, including changes in lighting, pose, facial expressions, and background environments.
Keywords: dataset, computer vision, face detection, annotation, normalization, augmentation, artificial intelligence.

ANNOTATSIYA: So'nggi yillarda sun'iy intellekt va kompyuter ko'rish texnologiyalarining rivojlanishi yuzni aniqlash masalalariga bo'lgan qiziqishni yanada oshirdi. Yuzni aniqlash tizimlarining samaradorligi ko'p jihatdan ularni o'qitishda foydalaniladigan dataset sifatiga bog'liq. Shu sababli, sifatli dataset tayyorlash ilmiy va amaliy jihatdan muhim vazifa hisoblanadi.
Mazkur maqolada yuzni aniqlash uchun dataset tayyorlash jarayoni batafsil yoritilgan. Jumladan, ma'lumotlarni yig'ish va ularni tozalash, yuzlarni belgilash (annotatsiya qilish), normalizatsiya va augmentatsiya bosqichlari tahlil qilinadi. Ushbu jarayonlar yordamida datasetning xilma-xilligi va balanslanganligi ta'minlanadi, bu esa o'z navbatida modelning aniqligi va barqaror ishlashini ta'minlaydi.
Tadqiqot shuni ko'rsatadiki, to'g'ri tayyorlangan dataset nafaqat yuzni aniqlash samaradorligini oshiradi, balki modelning turli sharoitlarda (yorug'lik, pozitsiya, ifoda, fon) ham yaxshi ishlashini kafolatlaydi.
Kalit so'zlar: dataset, kompyuter ko'rish, yuzni aniqlash, annotatsiya, normalizatsiya, augmentatsiya, sun'iy intellekt.

АННОТАЦИЯ: В последние годы развитие технологий искусственного интеллекта и компьютерного зрения усилило интерес к задачам распознавания лиц. Эффективность систем распознавания во многом зависит от качества набора данных, используемого при их обучении. Поэтому подготовка надежного и разнообразного набора данных является важной научной и практической задачей.
В данной статье подробно рассмотрен процесс подготовки набора данных для распознавания лиц. Анализируются этапы сбора и очистки данных, аннотации лиц, нормализации изображений и аугментации. Эти шаги позволяют обеспечить

разнообразие и сбалансированность набора данных, что напрямую влияет на точность и устойчивость работы моделей.
Исследование показывает, что правильно подготовленный набор данных повышает эффективность распознавания лиц и гарантирует надежную работу модели в различных условиях освещения, поз, выражений и фоновых элементов.

Ключевые слова: набор данных, компьютерное зрение, распознавание лиц, аннотация, нормализация, аугментация, искусственный интеллект.

1. Introduction

Currently, with the rapid development of information technology and artificial intelligence, the demand for face detection systems in computer vision has significantly increased [1]. The efficiency of face detection systems largely depends on the datasets they are trained on. Therefore, dataset preparation is considered one of the crucial stages in building such systems.
Dataset preparation involves collecting facial images, cleaning and annotating them, performing normalization, and applying artificial augmentation techniques [2]. However, this process is not always straightforward, as the collected data may contain various issues (e.g., low-quality images, incorrect labels, or duplicates). Hence, the use of specialized technologies for dataset preparation is required.
The scientific community has developed various tools to effectively organize the dataset preparation process. These tools can be divided into the following categories:
• Data collection tools: Google Images and OpenCV;
• Annotation technologies: VGG Image Annotator (VIA);
• Normalization and cleaning libraries: OpenCV, scikit-image, Pillow, imagehash;
• Augmentation tools: Keras ImageDataGenerator;
• Dataset management and visualization platforms: FiftyOne, DVC, Git LFS;
• Landmark and alignment technologies: MTCNN.

2. Google Images and OpenCV

Definition and Principles

Google Images is one of the largest image search engines in the world, indexing millions of pictures and graphic files across the internet. It allows users to quickly find images based on keywords, colors, image size, and other parameters. In dataset preparation, Google Images is often used during the data collection stage.

Google's web crawlers (search bots) index images from various websites and link them with relevant keywords. When a user submits a search query, the system returns results based on image descriptions, alt-tags, metadata, and page text.

For dataset creation, images can be automatically collected using the Google Custom Search API or scraping technologies (e.g., Python libraries such as BeautifulSoup, Selenium). However, it is very important to respect copyright and licensing conditions when using data collected from Google.

OpenCV is one of the most popular open-source libraries for computer vision, image and video processing, as well as machine learning. It supports both C++ and Python, and enables real-time image processing. In dataset preparation, OpenCV is often used for handling images and videos, face detection, normalization, and preprocessing.

OpenCV represents images as matrices (numpy arrays), meaning that each image is converted into a pixel array. It provides a wide set of ready-to-use functions for image processing, such as reading images, resizing, color conversion, applying blur or filters.

In OpenCV, face detection can be performed using the Haar Cascade classifier or Deep Neural Network (DNN) modules.

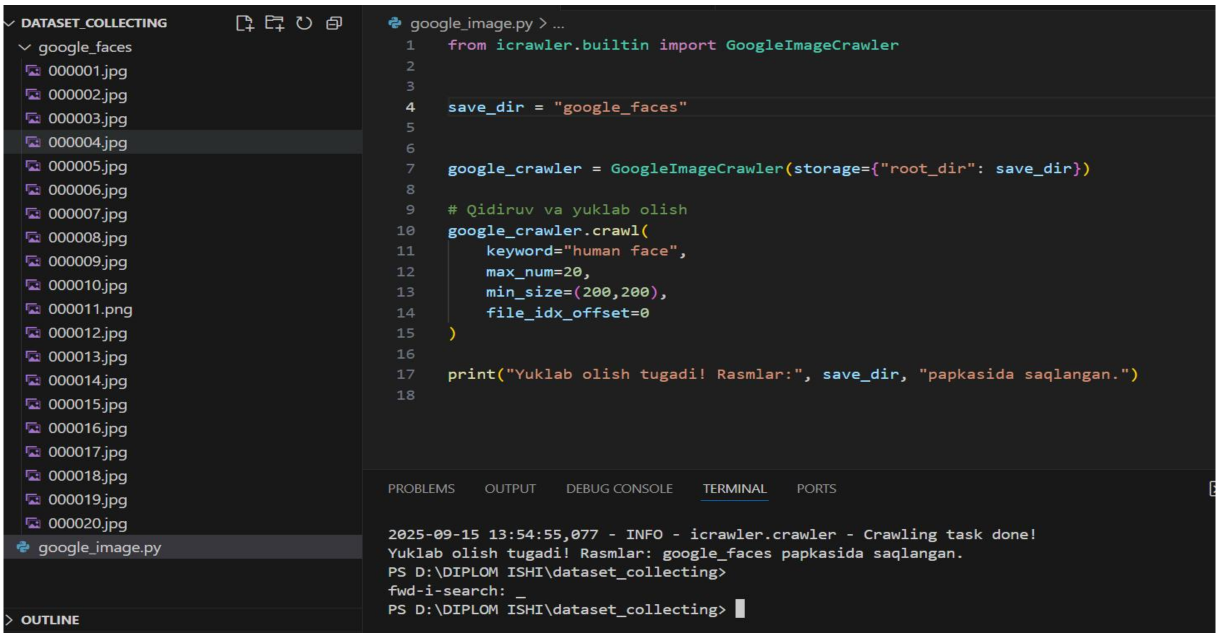
During dataset preparation, steps such as cropping faces, standardizing image sizes (resize), normalizing colors, and removing duplicates are performed using OpenCV.

2.2 Process

In the dataset preparation process, images are first collected using Google Images. Relevant keywords are selected for the search, and then Python libraries such as Selenium, BeautifulSoup, or the Google Custom Search API are used to automatically download the images. The downloaded files are checked for quality, and unclear or invalid images are removed. In the next stage, the OpenCV library is used to preprocess the collected images: resizing them to a standard size, detecting and cropping faces with the Haar Cascade or DNN module, standardizing colors, and filtering out very blurry or incorrect samples. In this way, images collected from Google Images are transformed into a clean and standardized dataset with the help of OpenCV tools.

ISSN: 2692-5206, Impact Factor: 12,23
American Academic publishers, volume 05, issue 09,2025

Journal: https://www.academicpublishers.org/journals/index.php/ijai



```python
from icrawler.builtin import GoogleImageCrawler


save_dir = "google_faces"


google_crawler = GoogleImageCrawler(storage={"root_dir": save_dir})

# Qidiruv va yuklab olish
google_crawler.crawl(
    keyword="human face",
    max_num=20,
    min_size=(200,200),
    file_idx_offset=0
)

print("Yuklab olish tugadi! Rasmlar:", save_dir, "papkasida saqlangan.")

```

```
2025-09-15 13:54:55,077 - INFO - icrawler.crawler - Crawling task done!
Yuklab olish tugadi! Rasmlar: google_faces papkasida saqlangan.
PS D:\DIPLOM ISHI\dataset_collecting>
fwd-i-search: _
PS D:\DIPLOM ISHI\dataset_collecting>
```

Figure 1. Downloading images from Google Images using Python

Using Python, we can download any images we want. As shown in the figure, we automatically downloaded 20 images using the keyword 'human face'.

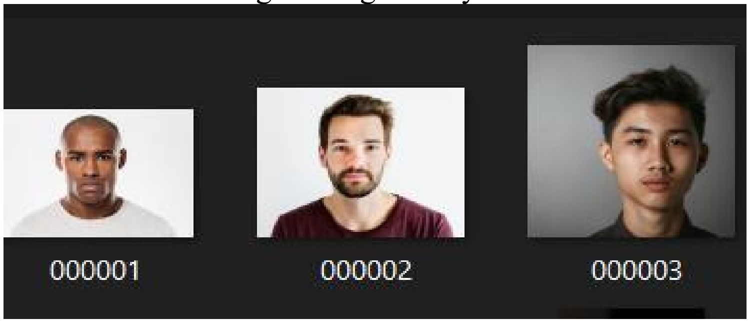


Figure 2. The samples from downloaded pictures

3. ANNOTATION TECHNOLOGIES

VGG Image Annotator (VIA) – is an open-source annotation tool developed by the Visual Geometry Group (VGG) at the University of Oxford. It works entirely in a browser and does not require installation. The user simply opens the *via.html* file and can immediately start the annotation process. Due to its lightweight, simple, and user-friendly design, VIA is widely used for annotating small and medium-sized datasets.

Types of annotation:

VIA provides users with the following types of labeling options:

• Bounding-box: marking a face or object within a rectangular frame.
• Polygon: accurately outlining complex shapes or face boundaries.
• Keypoints: marking specific coordinates such as eyes, nose, and mouth on a face.
• Circle/Ellipse: used for labeling certain special objects.

ISSN: 2692-5206, Impact Factor: 12,23
American Academic publishers, volume 05, issue 09,2025
Journal: https://www.academicpublishers.org/journals/index.php/ijai

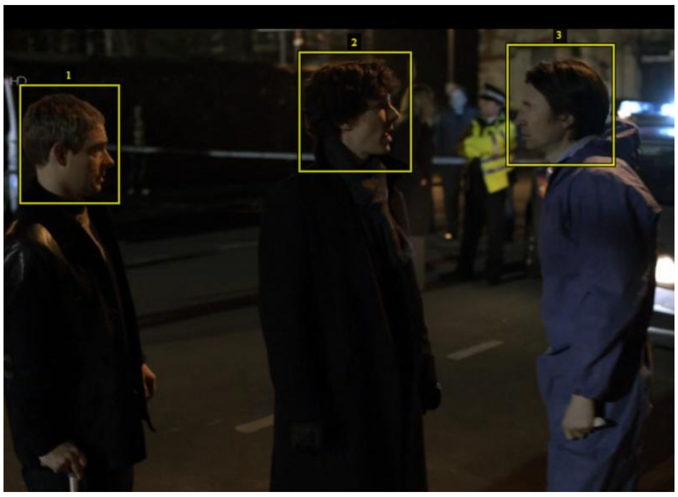


Figure 3. An image annotated with a bounding box using the VIA annotator.
Next, we will annotate the eyes, nose, and mouth using the VIA annotator. This will lead to very useful results during the training of our model.

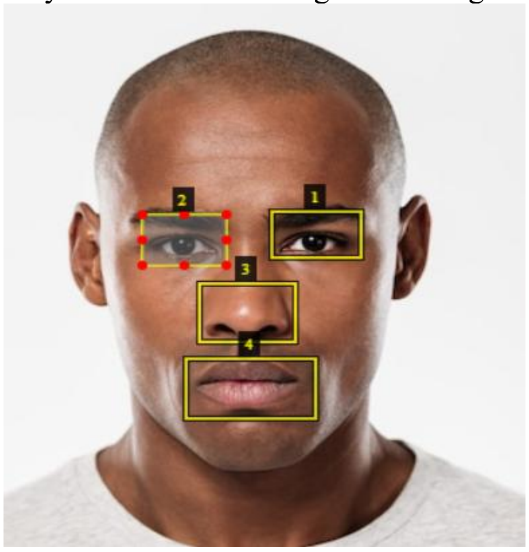


Figure 4. Annotated keypoints.
The VGG Image Annotator (VIA) is a lightweight, browser-based tool that supports multiple types of annotations and enables data export in formats such as JSON or CSV. In contrast, the Computer Vision Annotation Tool (CVAT) is a more advanced, professional platform designed for large-scale datasets and collaborative workflows, providing functionality for bounding-box, polygon, keypoint, and video annotations.

4. NORMALIZATION AND CLEANING LIBRARIES

Normalization and cleaning processes are crucial steps in improving the quality and efficiency of datasets prepared for face detection, as models may struggle with unnecessary noise, color variations, or images of inconsistent sizes. In such cases, Python libraries provide effective solutions. OpenCV enables image resizing, face cropping, and color standardization (e.g., converting from BGR to GRAY or RGB formats). Scikit-image offers functions for filtering, noise reduction, and contrast adjustment. Pillow (PIL) is widely used for basic image operations such as reading, rotating, cropping, and saving images. Meanwhile, the Imagehash library allows the detection and removal of duplicate or highly similar images within a dataset by applying hash functions to the images.

ORIGINAL ARTICLE

INTERNATIONAL JOURNAL OF ARTIFICIAL INTELLIGENCE

ISSN: 2692-5206, Impact Factor: 12,23

American Academic publishers, volume 05, issue 09,2025

Journal: https://www.academicpublishers.org/journals/index.php/ijai

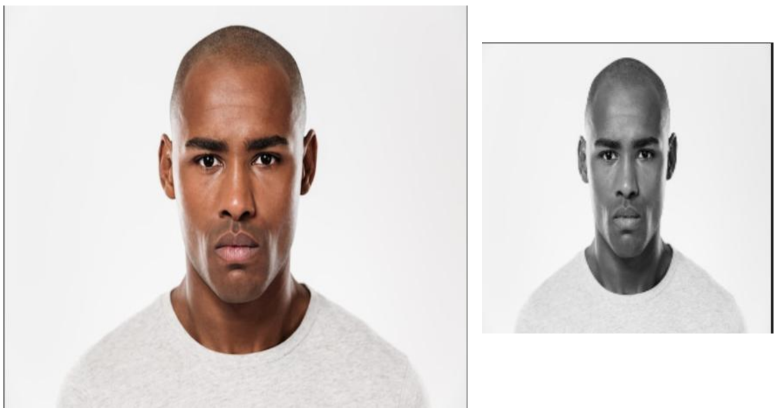


Figure 5. The original image has been converted to 224x224 size and grayscale format through annotation.

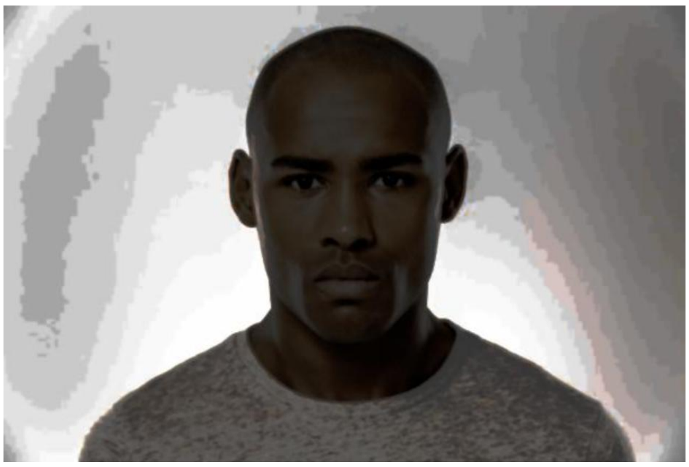


Figure 6. Image with adjusted contrast.

Contrast refers to the degree of difference between the bright and dark areas in an image. When the contrast is high, the bright areas appear very bright while the dark areas look very deep. In low contrast, however, the image appears "faded," and differences in brightness are less noticeable.

5. AUGMENTATION TOOLS

Augmentation refers to the process of artificially increasing the size of a dataset and improving the model's robustness to various conditions by applying different transformations to the images. This process is crucial in face recognition, as the model should not only perform well under a single type of lighting, rotation, or background condition, but also in diverse real-world scenarios. Through augmentation, the number of images is increased, and the generalization ability of the model is enhanced.

Common augmentation techniques used in practice include: image rotation, horizontal or vertical flipping, scaling (zooming in or out), adjusting color intensity (brightness, contrast, saturation), adding noise, random cropping, and geometric transformations such as shear or perspective transform.

In Python, several popular tools are available to implement these processes. OpenCV allows performing simple transformations such as rotation, flipping, and cropping. imgaug and Albumentations are specialized augmentation libraries that enable the application of complex combinations of transformations. KerasImageDataGenerator and TensorFlow's tf.image module are also widely used tools, providing real-time (on-the-fly) augmentation during model training. For PyTorch users, torchvision.transforms offers a convenient set of augmentation functions.

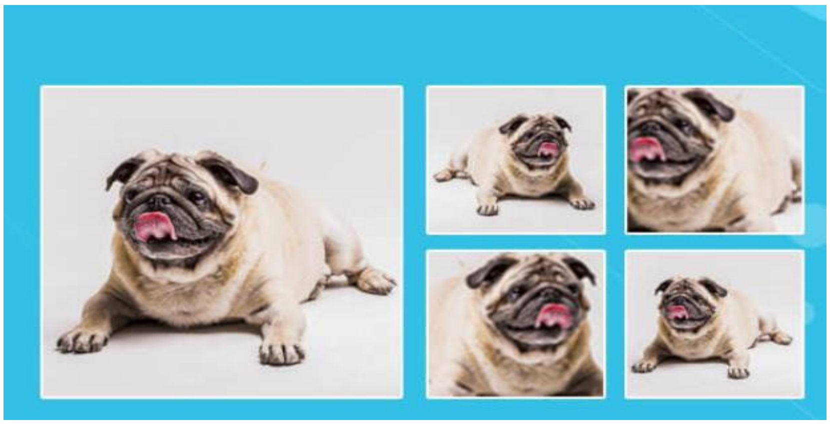


Figure 7. Image processed using Keras ImageDataGenerator.

Using the Keras ImageDataGenerator library, various transformations have been applied to the image. In this process, multiple new variants are generated from a single original image. ImageDataGenerator performs augmentation operations such as image rotation (rotation_range), zooming in or out (zoom_range), horizontal or vertical shifting (width_shift_range, height_shift_range), geometric transformations (shear_range), horizontal flipping (horizontal_flip), and many others. As a result, the dataset size is artificially increased, and the model is trained to be more robust under different conditions. As shown in the figure, the original image has been rotated, shifted, and zoomed in various ways, while preserving the semantic characteristics of the object. This contributes to strengthening the neural network's generalization capability.

6. LANDMARK AND ALIGNMENT TECHNOLOGIES

In the process of face detection and recognition, landmark and alignment technologies represent crucial steps. Landmark technology is used to detect key points on the human face, typically including coordinates of the eyes, nose, lips, and facial contours. Depending on the model, 5, 68, or 106 landmark points are commonly applied. One of the most well-known approaches for landmark detection is the shape_predictor model from the dlib library, which can accurately mark 68 key facial points. Using landmark technology, the geometric features of the face are modeled, enabling robustness under different conditions such as head rotation, facial expressions, or lighting variations.

Alignment, on the other hand, is the process of bringing the face into a "normalized position" using the detected landmark points. For example, if a person's face in an image is slightly tilted or turned sideways, alignment technology ensures that the eyes are aligned horizontally and the face is adjusted into a proper frontal view. The main goal of alignment is to reduce noise and improve accuracy during model training by standardizing the placement of facial components. From a technical perspective, alignment is often achieved through mathematical approaches such as Affine Transformation or Similarity Transformation, where facial coordinates are recalculated based on key landmarks (e.g., the centers of both eyes and the tip of the nose).

In practice, landmark and alignment technologies are applied together to standardize face images within a dataset, build models that are more robust to facial variations, and improve the performance of subsequent face detection or recognition algorithms.Stage 1: The Proposal Network (P-Net)

This first stage is a fully convolutional network (FCN). The difference between a CNN and a FCN is that a fully convolutional network does not use a dense layer as part of the architechture. This Proposal Network is used to obtain candidate windows and their bounding box regression vectors.
Bounding box regression is a popular technique to predict the localization of boxes when the goal is detecting an object of some pre-defined class, in this case faces. After obtaining the bounding box vectors, some refinement is done to combine overlapping regions. The final output of this stage is all candidate windows after refinement to downsize the volume of candidates.

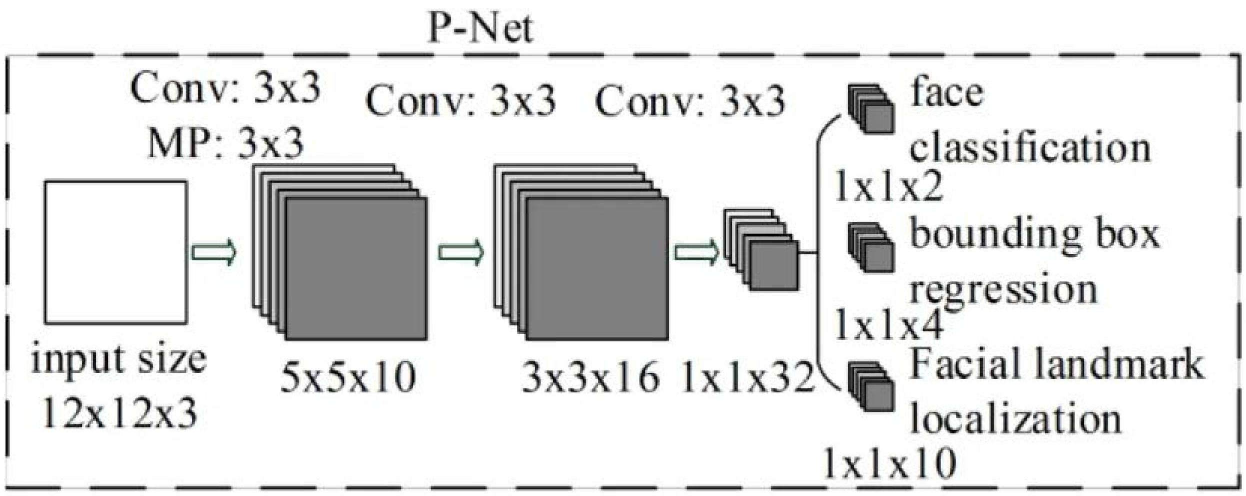


Figure 8. P-Net architecture.

Stage 2: The Refine Network (R-Net)

All candidates from the P-Net are fed into the Refine Network. Notice that this network is a CNN, not a FCN like the one before since there is a dense layer at the last stage of the network architecture. The R-Net further reduces the number of candidates, performs calibration with bounding box regression and employs non-maximum suppression (NMS) to merge overlapping candidates.
The R-Net outputs wether the input is a face or not, a 4 element vector which is the bounding box for the face, and a 10 element vector for facial landmark localization.

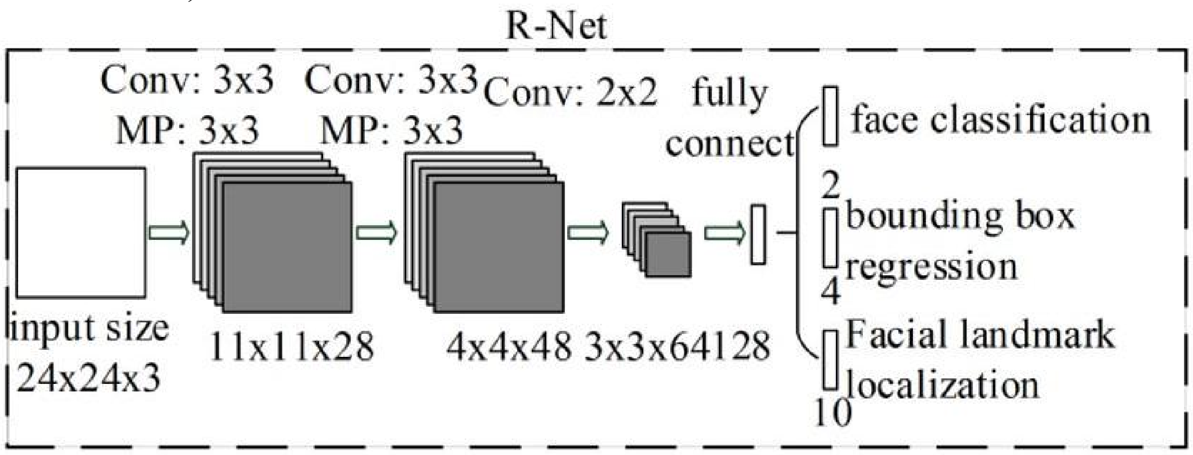


Figure 9. R-Net architecture.

Stage 3: The Output Network (O-Net)

ISSN: 2692-5206, Impact Factor: 12,23
American Academic publishers, volume 05, issue 09,2025

Journal: https://www.academicpublishers.org/journals/index.php/ijai

This stage is similar to the R-Net, but this Output Network aims to describe the face in more detail and output the five facial landmarks' positions for eyes, nose and mouth.

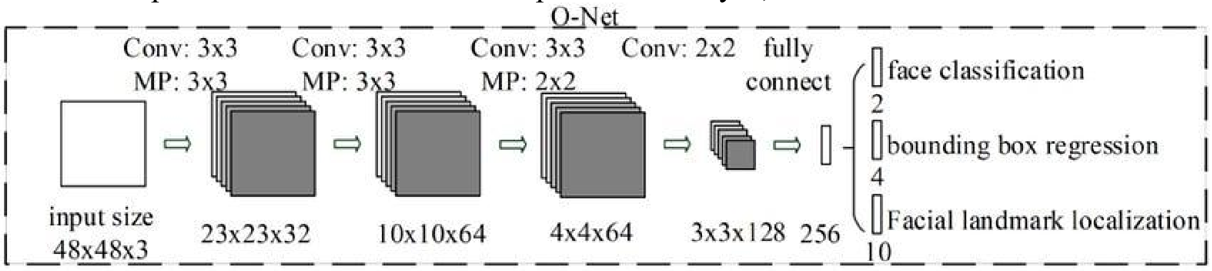


Figure 10. O-Net Architecture.

O-Net is the third and final stage of the MTCNN (Multi-task Cascaded Convolutional Neural Network) architecture, playing the most critical role in face detection and landmark localization tasks. The bounding-box candidates that pass through P-Net and R-Net are fed into O-Net, where the precise location of the face and its key landmarks are identified with high accuracy.

ISSN: 2692-5206, Impact Factor: 12,23
American Academic publishers, volume 05, issue 09,2025

Journal: https://www.academicpublishers.org/journals/index.php/ijai

CONCLUSION

Achieving high accuracy in the field of face detection requires not only efficient algorithms but also properly prepared and high-quality datasets. In this paper, we examined the dataset preparation process step by step: image collection, annotation, normalization and cleaning, augmentation, as well as the use of landmark and alignment technologies. While tools such as Google Images and OpenCV enable the collection of large-scale data, the VGG Image Annotator (VIA) provides convenience during the annotation process. Moreover, libraries such as OpenCV, scikit-image, and Pillow facilitate image standardization, cleaning, and quality enhancement.

Through the use of Keras ImageDataGenerator, it was demonstrated that dataset size can be artificially expanded, thereby improving the model's generalization capability. Landmark and alignment technologies, particularly MTCNN, ensure that key facial points are detected with high accuracy, resulting in more stable and complete training data for face recognition models.

Overall, properly prepared datasets significantly improve the accuracy and robustness of face detection and recognition systems. This creates a solid foundation for the effective application of face recognition technologies in various domains such as security, healthcare, social networks, and beyond. Research indicates that by combining modern technologies, the dataset preparation process can be further optimized, leading to a substantial increase in algorithm performance and efficiency.

ISSN: 2692-5206, Impact Factor: 12,23
American Academic publishers, volume 05, issue 09,2025

Journal: https://www.academicpublishers.org/journals/index.php/ijai

REFERENCES

1. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. IEEE Signal Processing Letters, 23(10), 1499–1503.
2. Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1, 511–518.
3. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition, 248–255.
4. Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The PASCAL Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, 88(2), 303–338.
5. Behnke, S. (2003). Hierarchical Neural Networks for Image Interpretation. Springer.
6. Chollet, F. (2015). Keras: Deep Learning library for Theano and TensorFlow. GitHub Repository. https://keras.io
7. Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.
8. Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). LabelMe: A database and web-based tool for image annotation. International Journal of Computer Vision, 77(1–3), 157–173.
9. Dutta, A., & Zisserman, A. (2019). The VIA Annotation Software for Images, Audio and Video. Proceedings of the 27th ACM International Conference on Multimedia, 2276–2279.
10. Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., & Yu, T. (2014). scikit-image: image processing in Python. PeerJ, 2, e453.